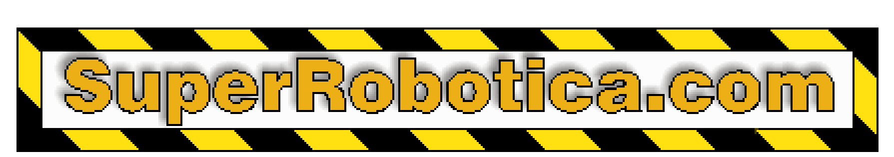


Basic Express
Guía del compilador

Versión 2.0



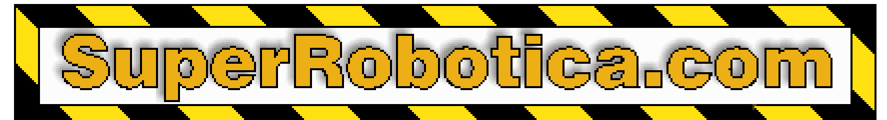




Traducción española: Alicia Bernal, Revisión: Pablo Pompa
www.superrobotica.com

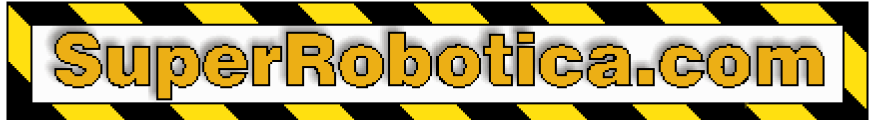
2.00.H





Contenido

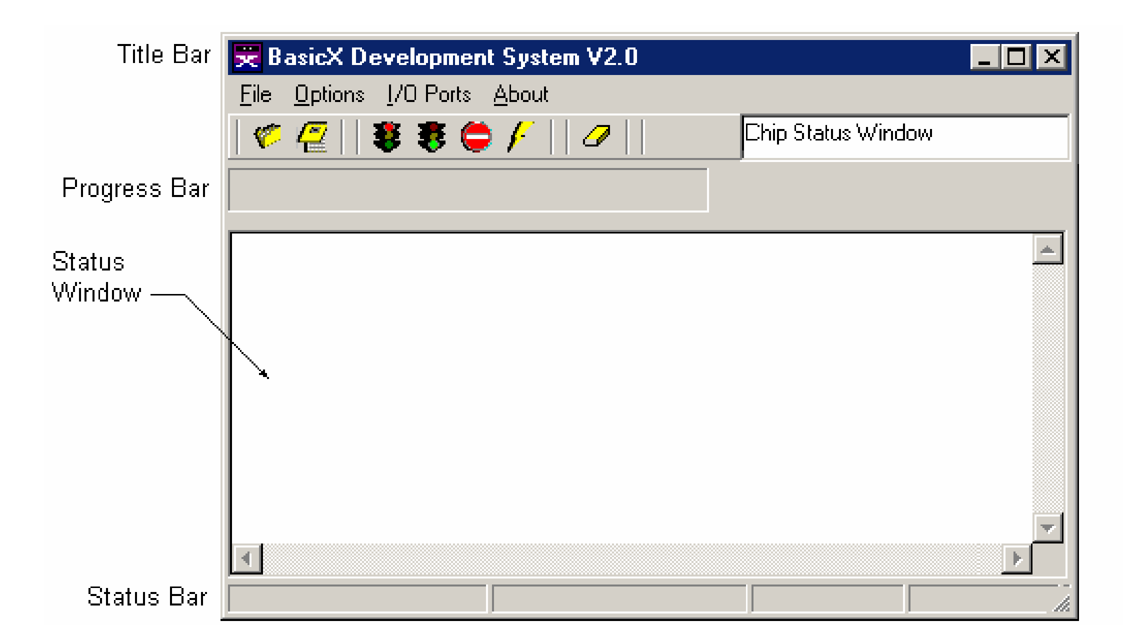




Programa de descarga

El programa de descarga de BasicX es donde se descargan los ficheros ejecutables y se ejecuta el sistema BasicX. Así mismo, también puede utilizar el programa de descarga para abrir el Editor/Compilador al crear o editar el proyecto BXP o los ficheros de código fuente BAS.

El programa de descarga muestra el resultado del proceso de descarga en la barra de estado. La ventana de estado permite una comunicación RS-232 bidireccional interactiva con el sistema BasicX.



Ventana del Programa de descarga

 **Botón Abrir fichero de descarga**

El botón Abrir fichero de descarga carga un fichero BXB y PRF en el sistema BasicX. El botón le pedirá que seleccione un nombre para el fichero BXB.

 **Botón Abrir Editor**

El botón del Editor abre el editor de BasicX. Aquí es donde se abre, edita y compila su proyecto.

 **Botón Detener Procesador**

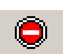
En los sistemas BX-01, el botón Detener Procesador corta la alimentación procedente del puerto paralelo. Este botón es útil cuando se cambian los componentes de la placa de desarrollo.

Advertencia – este botón no tiene efecto sobre la alimentación externa (conector de alimentación) de la placa de desarrollo.

En los sistemas BX-24 y BX-35, este botón detiene el procesador pero no tiene efecto sobre la alimentación.

 **Botón Ejecutar**

El botón Ejecutar ordena al programa BasicX que arranque y empiece a ejecutarse. Este estado se indica en la barra de estado con el mensaje RUNNING. El botón Ejecutar tiene el mismo efecto que liberar físicamente el botón de reinicio (Reset) de la placa de desarrollo.

 **Botón Reiniciar Procesador**

El botón Reiniciar Procesador detiene el programa BasicX. Este estado se indica en la barra de estado con el mensaje RESET. El botón tiene el mismo efecto que presionar físicamente el botón de reinicio (Reset) en la placa de desarrollo, aunque el botón del software tiene prioridad sobre el botón físico.

 **Botón Descargar Programa**

El botón Descargar Programa carga un fichero BXB en el sistema BasicX.

 **Botón Borrar Ventana Puerto Serie**

El botón Borrar Ventana Puerto Serie borra el contenido de la ventana de estado.

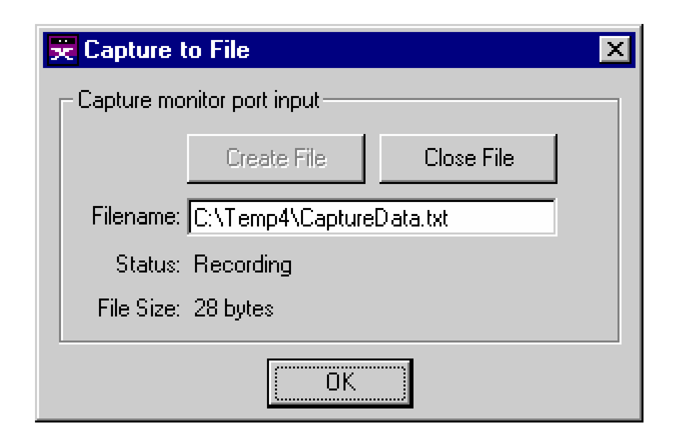
Menús del programa de descarga

Menú *File*

El menú *File* le permite abrir ficheros BXB y PRF, o salir del entorno de desarrollo BasicX (*BasicX Development Environment)*. Los ficheros BXB son ficheros binarios ejecutables de BasicX creados por el compilador, mientras que los ficheros PRF contienen la información adicional necesaria para el fichero BXB.

La opción *Open Download* abre los ficheros BXB y PRF especificados por el usuario. Después de abrir los ficheros, utilizará el botón de descarga para cargar el programa en el sistema de desarrollo.

La opción *Set Starting Directory* le permite especificar el directorio en el que el programa empieza a buscar los ficheros del proyecto que se desean abrir.



Ventana Capture to File

La opción del menú *Capture to File* le permite grabar los datos recibidos a través del puerto del monitor del PC. Los datos se almacenan en un fichero definido por el usuario, cuya longitud está tan sólo limitada por el espacio disponible en el disco. La grabación continúa hasta que se cierra el fichero o se salga del programa.

Menú del procesador (Processor Menu)

El menú Processor le permite seleccionar que tipo de sistema BasicX se va a utilizar. Las opciones son BX-01, BX-24 y BX-35.

Menú de puertos de entrada/salida (I/O Ports Menu)

El menú de puertos de entrada/salida (I/O Ports) le permite configurar los puertos de comunicación. El puerto de descarga (*Download Por)t* configura el puerto utilizado para la descarga de programas en el sistema BasicX. El puerto de descarga puede ser tanto un puerto paralelo como un puerto serie, dependiendo del sistema BasicX que esté utilizando.

Si está utilizando un puerto serie para la descarga, seleccione sólo el número de puerto Com. Otras configuraciones como la tasa de baudios y la paridad son fijas, por lo que el usuario no puede modificarlas.

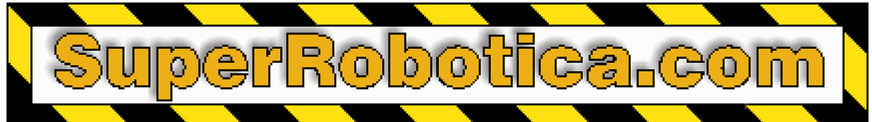
La opción *Monitor Port* configura el puerto serie que está conectado a la ventana de estado. Aquí puede especificar el número de puerto Com del PC, la tasa de baudios, paridad, número de bits de datos y número de bits de parada. Tenga en cuenta que el puerto del monitor es bidireccional – los datos recibidos de un sistema BasicX aparecerán en la ventana de estado, y los datos que escriba en la ventana de estado se transmitirán al sistema BasicX.

Tenga en cuenta que para los sistemas que hacen la descarga por el puerto serie, el puerto del monitor puede ser el mismo que el puerto de descarga. El programa conmuta los dos modos automáticamente.

La función *Rescue* (sólo para sistemas BX-24 y BX-35) obliga al procesador a aceptar un nuevo programa (ver página 7 para más detalles).

Barra de progreso

La barra de progreso muestra el porcentaje de descarga de su programa realizado en el sistema BasicX.

Ventana de estado
La ventana de estado muestra los datos serie recibidos del sistema BasicX a través del puerto de monitor del PC. Esta ventana también admite la entrada desde el teclado del PC para la transmisión serie al sistema BasicX.

Advertencia – la ventana de estado está pensada para comunicaciones relativamente sencillas con BasicX. La ventana no incorpora ningún protocolo de sincronismo cuando se comunica con un puerto serie de BasicX. En la práctica, esto no supone ningún problema a menos que BasicX esté enviando datos a una tasa muy elevada y el PC está ocupado con otras tareas de procesado.

En algunos casos la ventana de estado es capaz de detectar los errores de comunicación, tales como el desbordamiento del buffer de entrada. Si se detecta un error, aparecerá el mensaje "Serial Data Error Detected" (Se ha detectado error en los datos serie) en la parte izquierda de la barra de progreso.

Barra de estado

La barra de estado indica el estado de los distintos elementos del sistema. Entre estos se incluye estado encendido/apagado (on/off), estado del procesador (running/reset) y tamaño del programa.

Barra de título

La barra de título indica el número de versión del sistema de la placa de desarrollo BasicX.

Función Rescue (sólo BX-24 y BX-35)

La función Rescue es una opción que le permite controlar un procesador en funcionamiento y obligarle a que acepte un nuevo programa. La función Rescue es necesaria, ya que un programa BX-24 o BX-35 puede entrar en modos de funcionamiento que impidan que el sistema pueda aceptar un nuevo programa, incluso si reinicia de manera manual el sistema o desactiva la alimentación.

El sistema operativo BasicX debe estar conectado al puerto serie para poder aceptar un nuevo programa. Cualquier acción que interfiera con esto, como por ejemplo desactivar las interrupciones durante un periodo determinado de tiempo, podría causar problemas. Las llamadas del sistema PulseOut y PulseIn, por ejemplo, desactivan las interrupciones de manera temporal cuando se están ejecutando. Un programa que pase mucho tiempo llamando a estas funciones o similares podría necesitar la función Rescue.

La función Rescue permite al chip comunicarse dentro de los 0,25 segundos después de pulsar el botón de reinicio. La función Rescue detiene al procesador de una manera especial para que se pueda comunicar con el programa de descarga independientemente del tipo de programa introducido en el chip.

Procedimiento de recuperación:

1. Haga clic en la función Rescue

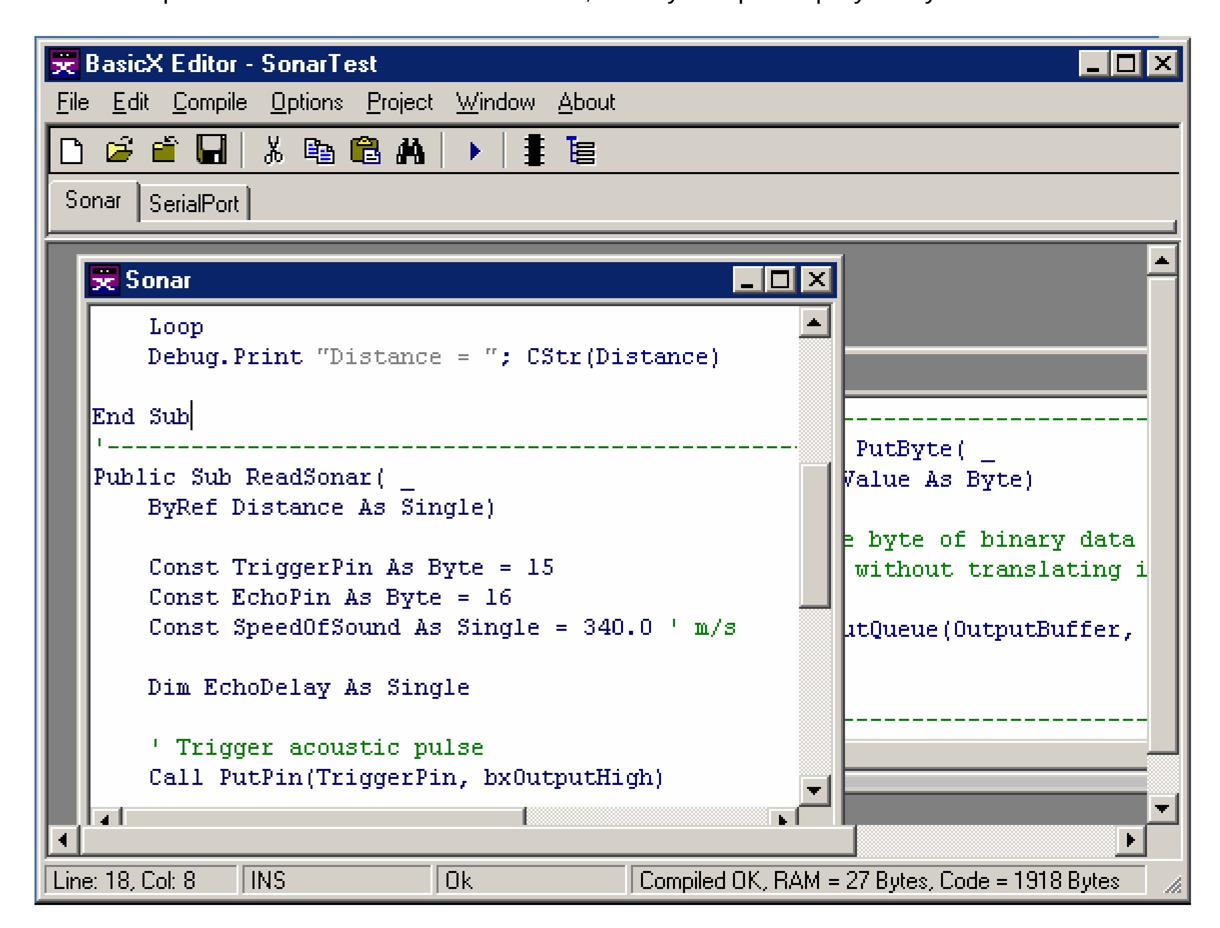
2. A lo largo de los 10 segundos siguientes, presione y libere el botón de reinicio en el procesador.

3. Después de ese periodo de 10 segundos, el sistema deberá estar listo para una nueva descarga.



Editor y compilador

El editor/compilador de BasicX en donde se abre, edita y compila el proyecto y sus ficheros.



Ventana del Editor

Barra de título y pestañas

La barra de título muestra el nombre del proyecto abierto actualmente. Las pestañas y las ventanas anidadas muestran los nombres de los módulos que se están editando.

Menú File

El menú *File* abre y guarda el proyecto y los ficheros de los módulos. Cada fichero de proyecto BXP registra el grupo de módulos que conforman cada programa.

Menú Edit

El menú *Edit* permite realizar varias operaciones de edición, como cortar (*cut*), copiar (*copy)*, pegar (*paste*), buscar y reemplazar (*search and replace*). Al hacer clic con el ratón derecho también podrá visualizar un menú desplegable con distintas opciones de edición.

Menú Compile

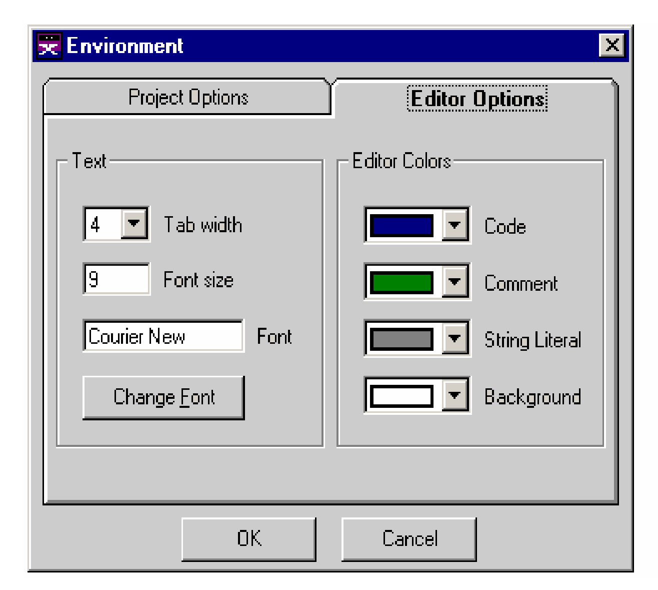
EL menú *Compile* tiene distintas opciones para compilar y ejecutar un proyecto. La opción *Compile* crea un fichero BXB ejecutable. La opción *Compile and Run* crea el fichero, lo descarga en el sistema BasicX e inicia el programa. La opción *Compile Status* muestra el estado de la última compilación, incluyendo los mensajes de error del compilador, si los hubiera.

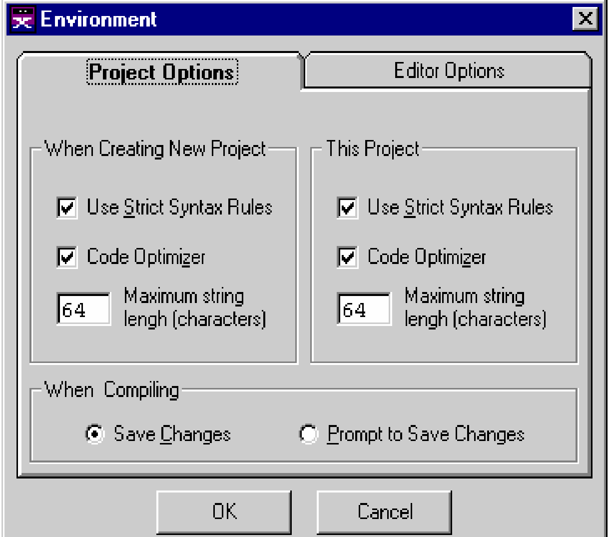
Menú Options

En el menú *Options*, la selección *Processor* (procesador) le permite seleccionar el tipo de sistema BasicX que va a utilizar. Las opciones son BX-01, BX-24 y BX-35 (esta opción del menú se trata de un duplicado de una de las ventanas del programa de descarga (downloader).

Si selecciona la opción *Environment* se abrirá la ventana que aparece a continuación que le permite controlar varias especificaciones del entorno de desarrollo.

Las distintas configuraciones del editor están disponibles en la pestaña *Editor Options (Opciones del editor)*, incluyendo el ancho de la pestaña, tamaño y tipo de fuente. También es posible destacar el texto a través de distintos colores definidos por el usuario para los códigos, comentarios y literales de cadenas.





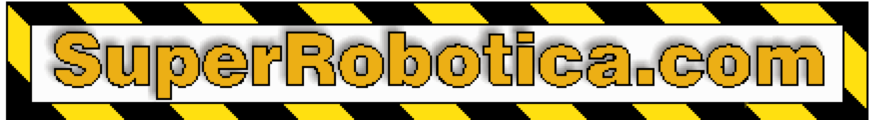
Ventana Environment

En la pestaña *Project Options* (Opciones del proyecto) puede especificar que los cambios realizados se guarden automáticamente en el disco cuando compile un programa. También puede hacer que el compilador le pregunte si quiere guardar los cambios.

Así mismo, puede configurar los valores de las tres opciones del compilador:

Code optimizer (Optimizador de códigos)

Si selecciona esta casilla de control podrá controlar si el compilador optimiza el programa o no. Normalmente, la optimización reduce el tamaño de un programa.

Strict syntax rules (Reglas sintácticas estrictas)

Esta casilla de control impone varias restricciones en la codificación para que el código sea más fácil de leer y comprender. Las restricciones afectan a los contadores de bucles for-next, literales numéricos y operaciones lógicas a bits. Consulte el manual de lenguaje si desea obtener más detalles.

Maximum string length (Longitud máxima de cadenas)

Este cuadro de texto le permite especificar la longitud máxima permitida de las cadenas de longitud variable o fija. El rango de selección es 1 - 64 caracteres. EL valor por defecto es 16 caracteres.

Dado que las cadenas de longitud variable requieren el mismo espacio de almacenamiento, puede reducir los requisitos generales de almacenamiento seleccionando un valor óptimo para el tamaño de las cadenas.

Menú Project

El menú *Project* tiene distintas opciones que afectan al proyecto. La opción *Chip* abre el cuadro de diálogo Chip en el que se incluyen opciones para hacer más fáciles las rutinas de configuración, como por ejemplo definir los estados iniciales de los pines de entrada/salida (I/O). La opción *Files* abre la ventana de diálogo Project, que le permite añadir o eliminar módulos del proyecto. La ventana *Watch Window* le permite supervisar las variables de un programa BX-24 o BX-35 en ejecución.

Menú Window

El menú *Window* le permite especificar la forma de visualización de las ventanas anidadas en el editor. Las opciones son *Cascade* (Cascada), *Tile Horizontal* (Mosaico horizontal) y *Tile Vertical* (Mosaico vertical).

Edit Window

La opción *Edit Window* (Ventana de edición) es donde se crea y edita el código para los ficheros de los módulos. En la parte inferior de esta ventana de edición se encuentra la Barra de estado (Status Bar)



Barra de estado

La barra de estado indica el estado de los distintos elementos del sistema.

El panel 1 indica el número de línea y columna del cursor de texto. El panel 2 indica si el cursor de texto está en modo insertar (INS) o sobre escribir (OVR).

El panel 3 indica si los errores o advertencias has sido generados por el compilador. El panel 4 muestra si la compilación se ha realizado con éxito o no.

Si la compilación se ha realizado correctamente (OK), en el panel 4 aparecerá la cantidad de memoria EEPROM utilizada por el código (Code), la memoria RAM asignada a las variables estáticas,

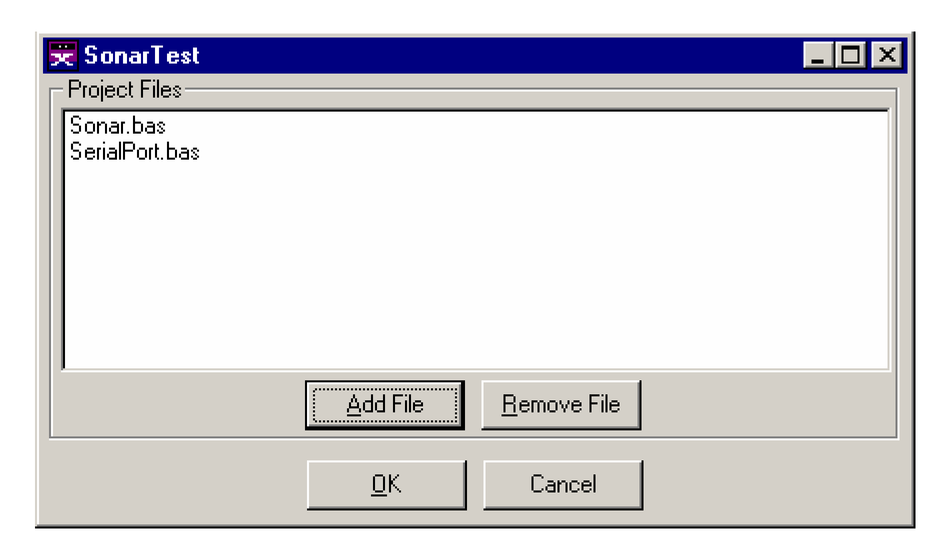
y la memoria no volátil asignada a las variables persistentes (abreviado como PRS). Por otro lado, si la compilación no se ha realizado con éxito, en el panel aparecerá un mensaje de error fatal.

La información descrita anteriormente puede también visualizarse en una ventana independiente haciendo clic en la barra de estado o seleccionado *Compile Status* en el menú *Compile*. Puede también ver más detalles relacionados con la asignación de memoria consultando el fichero de mapa MPP, creado al compilar un programa.

Advertencia – la memoria RAM que aparece en la barra de estado se refiere únicamente a la memoria asignada a las variables estáticas. Esto no equivale a la cantidad total de RAM que requiere un programa. En concreto, las pilas de tareas requieren RAM adicional, que no aparecen aquí.

Cuadro de diálogo Project

El cuadro de diálogo Project registra el código fuente del proyecto. El código fuente se almacena en un o varios ficheros de módulos, cuya extensión por defecto es .BAS. El grupo de módulos se compila en el fichero final ejecutable con formato .BXB.



Ventana de diálogo Project

Barra de título

En la barra de título aparece el nombre del fichero BXP del proyecto abierto actualmente.

Ficheros de proyecto

En la caja de texto *Project Files* (ficheros del proyecto) aparecen enumerados todos los ficheros del proyecto. Cada fichero contiene el código fuente para un solo módulo. Puede seleccionar qué ficheros va a incluir pulsando los botones *Add* / *Remove File* para añadir / quitar ficheros.

Añadir fichero

El botón *Add File* (Añadir fichero) abre un cuadro de diálogo en el que puede seleccionar los ficheros de los módulos que puede añadir al proyecto actual.



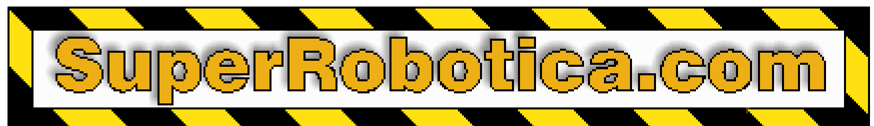
Quitar fichero

El botón *Remove File* (Quitar fichero) elimina el fichero destacado del proyecto.

Botón OK

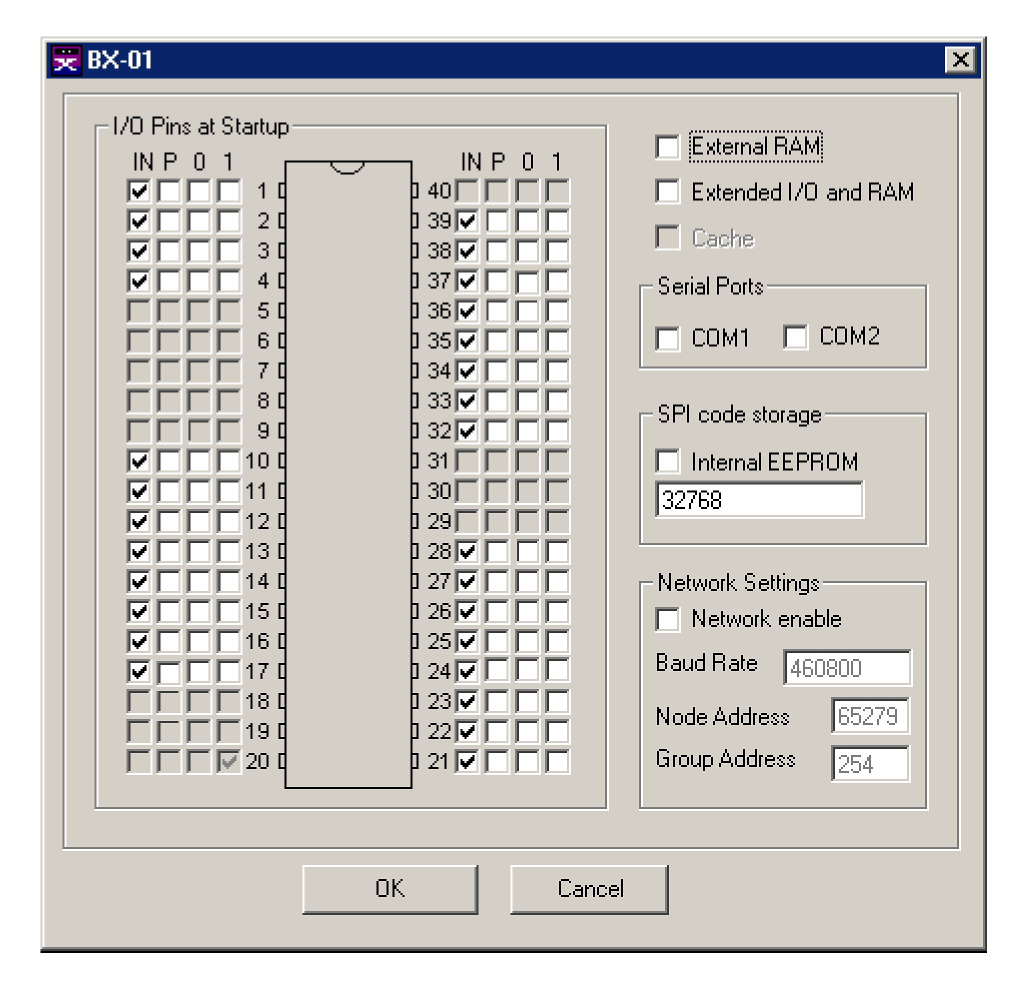
El botón OK (Aceptar) sale del cuadro de diálogo Project después de guardar todos los cambios realizados dentro de ese cuadro de diálogo.





Cuadro de diálogo del chip de BX-01

El cuadro de diálogo del Chip le permite controlar las opciones del chip del sistema BasicX, como la RAM externa, los puertos serie COM1/COM2, SPI interna, tamaño de SPI y entradas/salidas (I/O) ampliadas. Puede configurar las opciones de red como la tasa de baudios, y la dirección de nodo/grupo. Así mismo, puede controlar el estado inicial de cada pin de entrada/salida. (Tenga en cuenta que algunas de estas opciones pueden estar controladas por el programa BasicX por si mismo).



RAM externa

La opción *External RAM* configura el BasicX para que utilice la memoria RAM adicional del RamSandwich™. La memoria RAM externa requiere los pines 2, 16, 17, 21-28, y 32-39.

Cache

La opción *Cache* puede activarse al utilizar la memoria RAM externa. Una vez seleccionada esta casilla de control, cargará su programa en la memoria donde se ejecutará más rápido.

COM1

La opción COM1 configura a BasicX para que utilice el dispositivo COM1 integrado. Esta opción deshabilita la red. COM1 requiere los pines 10 y 11.

COM2

La opción COM2 configura a BasicX para que utilice el dispositivo COM2 integrado, que requiere los pines 1 y 12.

Entradas / Salidas adicionales y memoria (RAM Extended I/O and RAM)

Esta opción configura a BasicX para que utilice un dispositivo externo como Video-X™. Las entradas/salidas (I/O) adicionales requiere los pines 3, 21-28, y 32-39.

Almacenamiento de código SPI (SPI code storage)

El almacenamiento del código SPI define el espacio máximo para el programa disponible en el chip. El compilador realiza comprobaciones para garantizar que su programa encaja dentro de la memoria permitida.

Memoria EEPROM interna (Internal EEPROM)

La opción de memoria EEPROM interna determina si su programa se descargará internamente en el chip de BasicX, o externamente en otro chip SPI. SPI externo requiere los pines 6, 7, y 8. Si selecciona esta casilla de control selecciona el SPI interna y libera esos 3 pines.

Opción Habilitar red (Network Enable)

La opción Network Enable (Habilitar red) configura a BasicX para que utilice la capacidad integrada de red. La red requiere el dispositivo COM1 y los pines 10, 11, y 14.

Opción Dirección de nodo (Node Address)

La opción Node Address (Dirección de nodo) establece el nodo de dirección de red para el procesador. Cada chip de una red particular debe tener una dirección única. Las direcciones de nodos válidas se encuentran en el rango de 0 a 65 279.

Dirección de grupo (Group Address)

La opción de Dirección de grupo le permite definir la dirección del grupo del chip. Es posible utilizar múltiples nodos en una red que compartan la misma dirección de grupo. La opción *groupcasting* le permite enviar un paquete de red de manera simultánea para todos los miembros de un grupo determinado. Las direcciones de grupo válidas se encuentran en el rango de 0 a 254.

Tasa de baudios (Baud Rate)

La opción de tasa de baudios configura la tasa de comunicación de la red. La tasa máxima (y el valor por defecto) es 460.800 baudios.

Botón OK

El botón OK sale de la ventana después de guardar cualquier cambio realizado en el cuadro de diálogo del Chip.

Botón Cancel

El botón Cancel sale de la ventana, ignorando cualquier cambio realizado en el cuadro de diálogo del Chip.

Cuadrículas de los pines de entrada/salida (I/O)

Las cuadrículas de los pines I/O le permite definir el estado inicial de cada pin de entrada/salida disponible siempre que se reinicia el procesador. Cada pin puede configurarse en uno de los cuatro estados posibles -- tristate, pullup, output-high y output-low (triestado, polarización positiva, salida nivel alto y salida nivel bajo)

IN **Input--** Configura los pines correspondientes como una entrada de modo tristate (alta impedancia). Normalmente 5 V se considera como un nivel lógico alto (en sistemas de 5 V) mientras que 0 V se considera un nivel lógico bajo.

P **Input w/ Pullup --** Configura el pin como una entrada con polarización positiva. Este estado se utiliza normalmente para controlar el estado de los dispositivos pasivos como por ejemplo los interruptores.

0 **Output Low --** Configura el pin correspondiente como salida de nivel lógico bajo, que es 0 V.

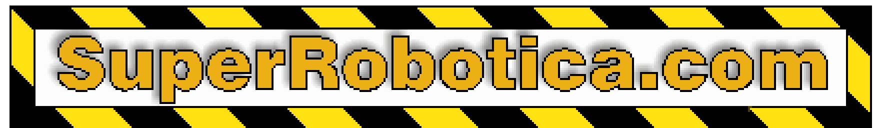
1 **Output High --** Configura el pin como una salida de nivel lógico alto, que normalmente es 5 V.

Las cuadrículas de pines I/O definen sólo los estados iniciales de cada uno de los pines. Después del arranque, puede también hacer que el programa configure cada pin independientemente utilizando el procedimiento PutPin u otras llamadas del sistema.

Errores conocidos

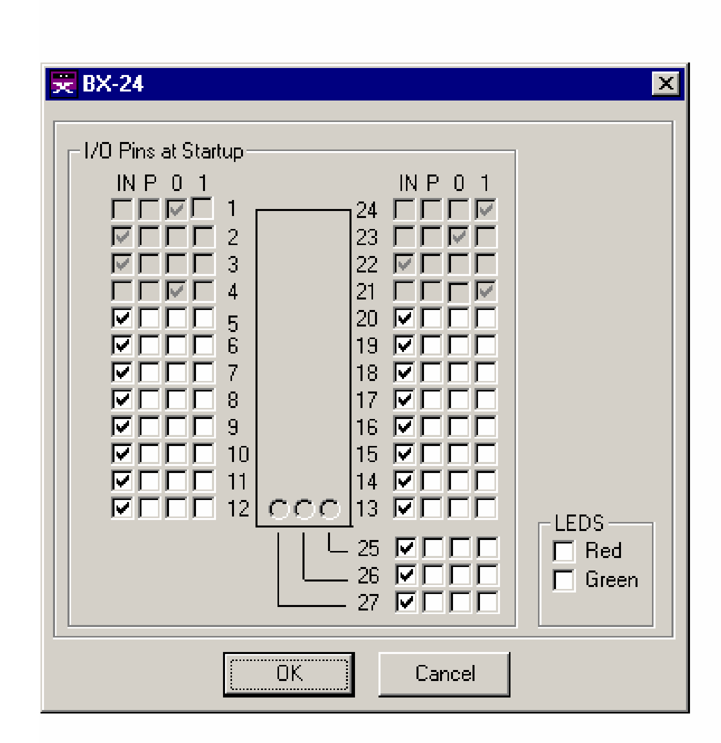
Si activa la opción cache, es posible que en algunos casos el programa no arranque después de la descarga o del reinicio. Hay un periodo de retardo para desactivar esta opción.

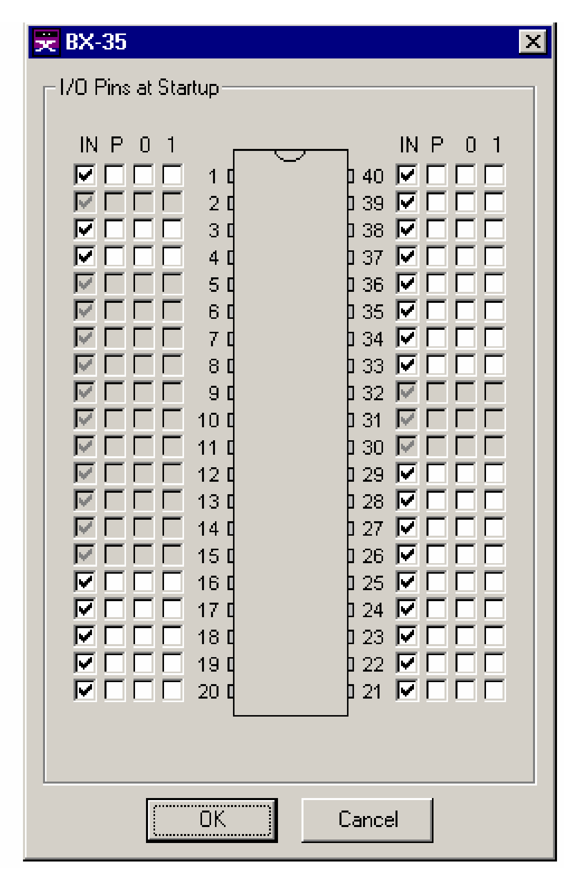




Cuadros de diálogos de chip BX-24 y BX-35

Los cuadros de diálogo del chip BX-24 y BX-35 le permiten controlar el estado inicial de los pines I/O y los LEDs en placa de un sistema BX-24. Estas opciones puede también controlarse desde el programa BasicX.





Cuadrículas de piles de entrada/salida (I/O)

Las cuadrículas de pines de entradas/salidas (I/O) le permite definir el estado inicial de cada pin I/O disponible cuando se reinicia el procesador. Cada pin puede configurarse con uno de los cuatro estados -- tristate, pullup, output-high y output-low (triestado, polarización positiva, salida nivel alto y salida nivel bajo):

IN **Input--** Configura los pines correspondientes como una entrada de modo triestado (alta impedancia). Normalmente 5 V se considera como un nivel lógico alto (en sistemas de 5 V) mientras que 0 V se considera un nivel lógico bajo.

P **Input w/ Pullup --** Configura el pin como una entrada con polarización positiva. Este estado se utiliza normalmente para percibir el estado de los dispositivos pasivos como por ejemplo los interruptores.

0 **Output Low --** Configura el pin correspondiente como salida de nivel lógico bajo, que es 0 V.

1 **Output High --** Configura el pin como una salida de nivel lógico alto, que normalmente es 5 V.

Las cuadrículas de pines I/O definen sólo los estados iniciales de cada uno de los pines. Después del arranque, puede también hacer que el programa configure cada pin independientemente utilizando el procedimiento PutPin u otras llamadas del sistema.

Botón OK

El botón OK sale de la ventana después de guardar cualquier cambio realizado en el cuadro de diálogo del Chip.

Botón Cancel

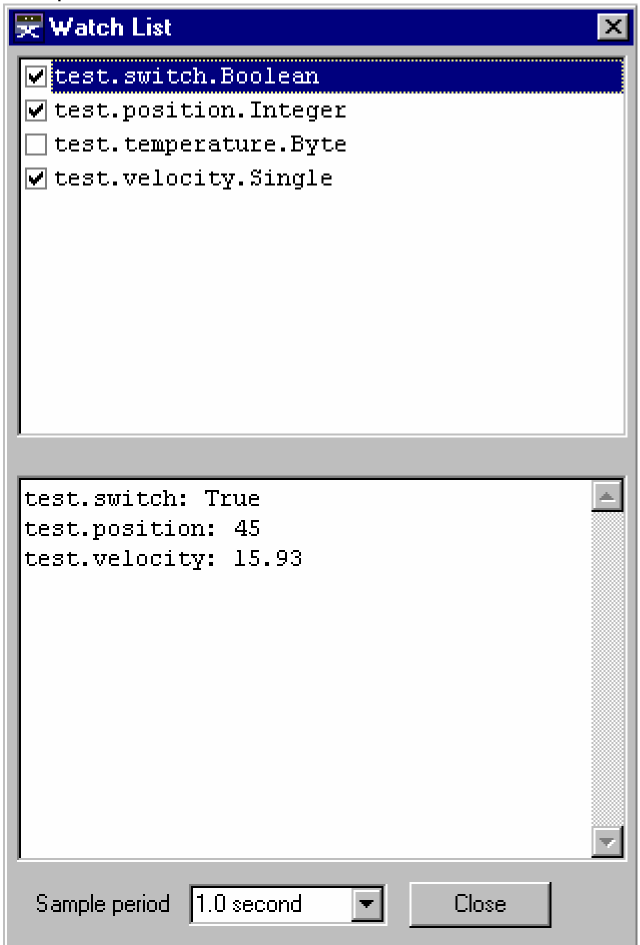
El botón *Cancel* sale de la ventana, ignorando cualquier cambio realizado en el cuadro de diálogo del Chip.

Casillas de control LEDS (sólo BX-24)

La casilla de control LEDS le permite definir el estado inicial de cada uno de los LEDs del sistema BX-24. Así mismo, puede hacer también que su programa controle el estado de cada uno de los LEDs a través de las funciones de los pines de entrada/salida (I/O) – el LED rojo está conectado al pin 25 mientras que el LED verde está conectado al pin 26. Para encender un LED, puede es necesario establecer el pin como nivel lógico bajo. Subir el nivel apagaría el LED.

Ventana Watch (sólo BX-24 y BX-35)

La ventana de inspección *Watch* le permite supervisar las variables de un programa BasicX en ejecución. La ventana *Watch* puede manejar variables estáticas (nivel de módulo) en la memoria RAM. Los tipos de las variables pueden ser numéricos o booleanos. Si se selecciona una matriz, el primer elemento de la matriz aparecerá en la ventana:



Ventana Watch

A la ventana *Watch* se accede a través del menú *Project* del editor.

En la ventana superior se enumeran las variables de nivel de módulo basadas en la memoria RAM del programa. Las casillas de control le permiten seleccionar el tipo de variables que desea inspeccionar. En la ventana inferior se muestran los valores actuales de las variables seleccionadas.

La ventana *Watch* revisa de manera periódica el procesador BasicX para actualizar cada variable. La caja combinada situada en la parte inferior de la ventana le permite personalizar la tasa de muestra seleccionando el periodo dentro del rango de 1 a 5 segundos.

La ventana *Watch* lee el fichero BXM, que se genera al compilar un programa. Consulte la siguiente sección si desea ver más detalles acerca del fichero BXM.

Advertencia – la ventana *Watch* necesita el puerto serie Com1 del procesador BasicX para poder comunicarse. El programa no debería utilizar el puerto Com1 mientras la ventana *Watch* está activa, y el puerto del monitor debería estar cerrado.

Así mismo, cualquier cosa que interfiera las comunicaciones por el puerto serie, como por ejemplo apagar interrupciones, interferirá en la ventana *Watch*.

Las variables y objetos siguientes no se muestran en la ventana *Watch*:

Variables locales
Variables persistentes
Variables de cadena
Objetos de registro
Objetos de datos de bloque

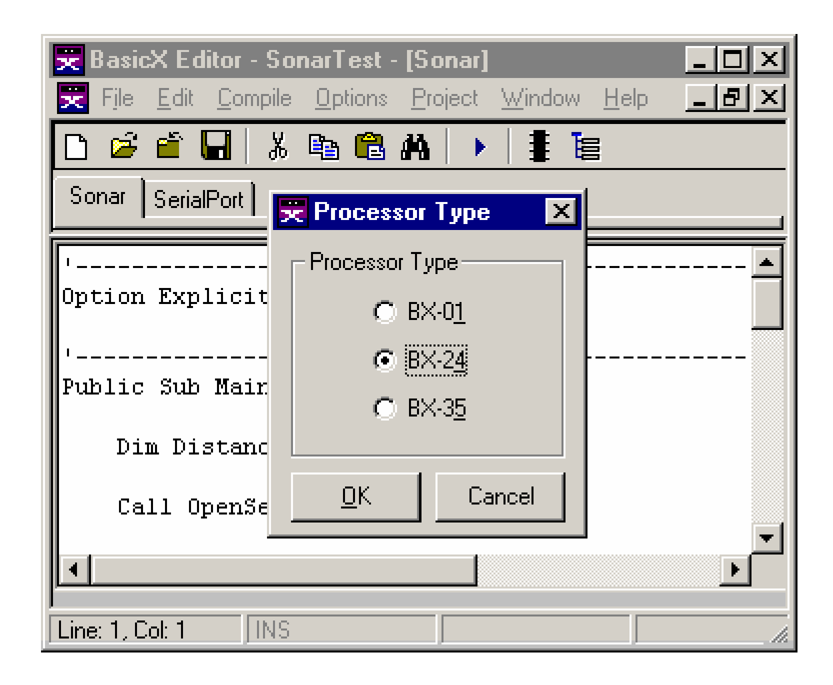
Tutorial sobre el uso de la ventana *Watch*

Procedimiento

Para ilustrar el funcionamiento de la ventana *Watch*, utilizaremos el siguiente programa como ejemplo. El objetivo es visualizar el valor de la variable estática **i** mientras el programa esté en ejecución:

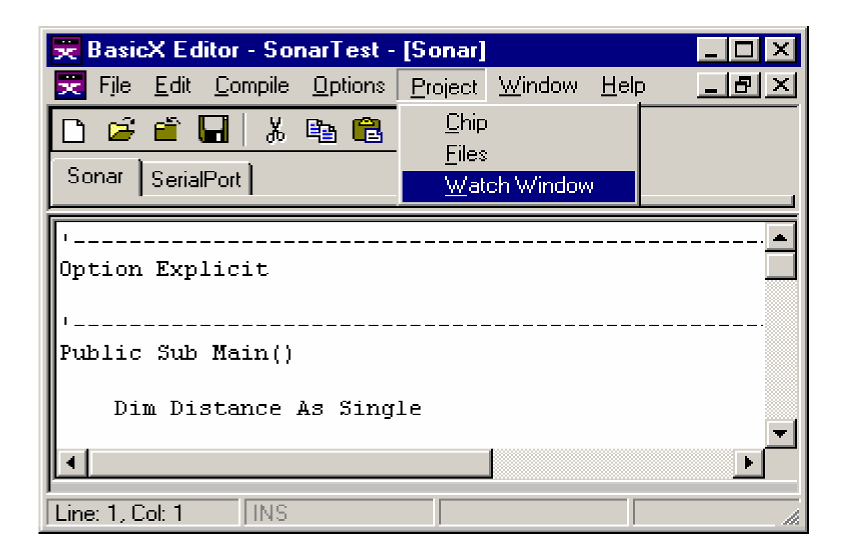
```
Dim i As Integer
Sub Main()
    i = 1
    Do
        i = i + 1
        If (i > 5) Then
            i = 1
        End If
        Call Delay(2.0)
    Loop
End Sub
```

Para iniciar la ventana Watch, el primer paso es abrir el editor y comprobar que se ha seleccionado el tipo de procesador BX-24 o BX-35 en el menú Options-Processor:

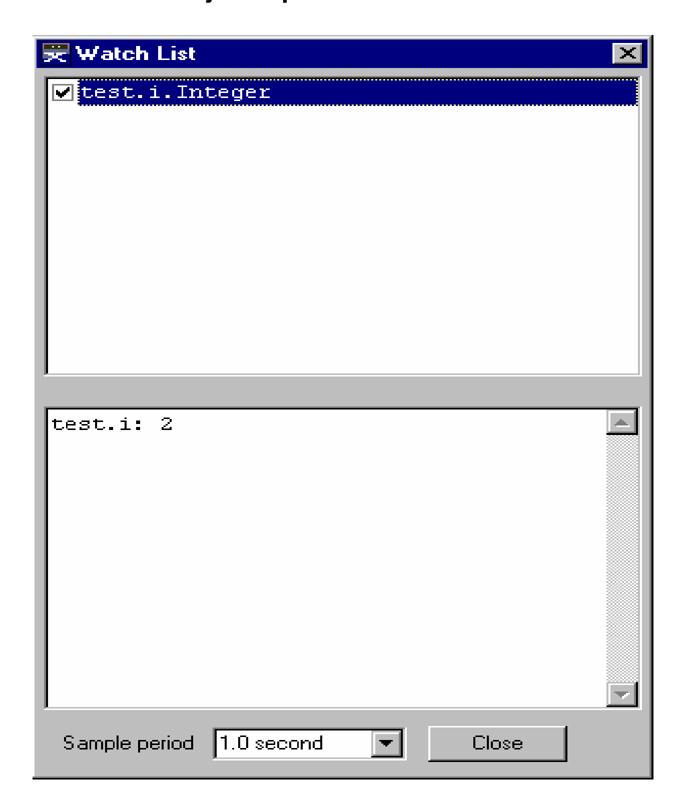


Ventana de tipo de procesador

El siguiente paso es compilar y descargar su programa como siempre. Una vez que empiece a ejecutarse el programa, seleccione el menú *Project* - *Watch Window*.



Menú Project-Opción Watch Window



Debería aparecer la ventana Watch. En la ventana superior, puede seleccionar una o varias variables de nivel de módulo que desea supervisar, y el compilador actualizará de manera dinámica los valores de cada variable en la ventana inferior. Tenga en cuenta que el tipo de variable aparece justo después de cada identificador en la ventana superior:

Ventana Watch



En este caso, la variable *test.i* se actualizará una vez por segundo. La tasa de actualización puede seleccionarse en la caja de lista situada en la parte inferior de la ventana.

Tipos de ficheros

El programa de compilación/descarga lee y genera los siguientes tipos de ficheros:

Extensión de ficheros	Descripción
BAS	Fichero de módulo -- contiene el código fuente para el programa.
BXP	Fichero de proyecto – detalla todos los módulos del programa.
PRF	Fichero de preferencias -- contiene la información de inicialización y configuración.
BXB	Programa BasicX en formato binario -- similar a un fichero EXE de un PC.
MPP	Fichero de mapa de todo el programa, en un formato legible para humanos.
BXM	Fichero de mapa de las variables de nivel de módulo, en un formato legible para máquinas.
MPX	Fichero de mapa de las variables de nivel de módulo, en formato de código fuente para la red.
OBJ	Fichero de objeto -- contiene el código objeto de cada subprograma.
ERR	Fichero de registro de error, producido por el modo de línea de comando.

Fichero BAS

Este fichero contiene el código fuente de un solo módulo. Cada módulo debe de estar en un fichero independiente, con uno y sólo un módulo por fichero. Las extensiones de los ficheros pueden ser arbitrarias, siendo BAS la extensión por defecto.

El nombre del módulo se toma del nombre del fichero sin la extensión. El nombre de módulo debe ser un identificador legal de Basic, por lo que el nombre debe empezar por una letra, y el resto de los caracteres deben ser letras, dígitos o guiones bajos.

Fichero BXP

Este es fichero de proyecto, que contiene una lista con los nombres de ficheros de todos los módulos del programa. Formato típico:

```
C:\Dir1\MainModule.bas
C:\Dir3\SubDir\Irrigation.bas
SerialPort.bas
```

Los nombres de los ficheros tienen el prefijo de rutas de acceso opcionales. Si no está presente la ruta de acceso, el compilador presupone que el fichero está ubicado en el mismo directorio que el fichero BXP.

Fichero PRF

Este es el fichero de preferencias, que contiene la información requerida por el sistema BasicX. Esta información incluye los estados iniciales de los pines I/O, así como los datos de la configuración de la memoria.

Es necesaria la combinación de un fichero PRF y BXB siempre que se descargue un programa en un sistema BasicX.

Fichero BXB



El fichero BXB contiene el código ejecutable que se ejecuta en el chip de BasicX. Este fichero, combinado con el fichero PRF, es similar al fichero EXE de un PC.

Fichero MPP
Este fichero contiene un mapa de todo el programa. Aparecen las siguientes categorías:

Variables de nivel de módulo en RAM
Variables persistentes (EEPROM)
Datos locales del subprograma
Memoria de códigos (subprogramas y datos de bloque)

Fichero BXM
El fichero BXM contiene una descripción de todas las variables (estáticas) de nivel de módulo y matrices del programa. El fichero está en un formato ASCII legible para máquinas. Cada línea del fichero contiene los siguientes datos en un formato delimitado por comas:

ModuleName.VariableName – El nombre del módulo y de la variable (o matriz) dentro de las comillas, que utilizan un separador de punto.

MemoryLocation – Dirección del inicio de la variable.

DataType – Enumeración de los tipos de datos (ver abajo).

IsPersistent – Si la variable es persistente. El valor es #TRUE# o #FALSE# (Verdadero/Falso)

Dimensions – Número de dimensiones. El escalar es cero.

Se ofrece una lista con el límite inferior y superior de cada una de las ocho dimensiones posibles. Para las variables de cadena, el primer límite superior es el número máximo de caracteres que pueden almacenarse en la cadena.

Enumeraciones de los tipos de datos:

Tipo	**Valor**
UnsignedInteger	17
Byte	18
Integer	19
Long	20
UnsignedLong	21
Single	32
Boolean	50
String	61

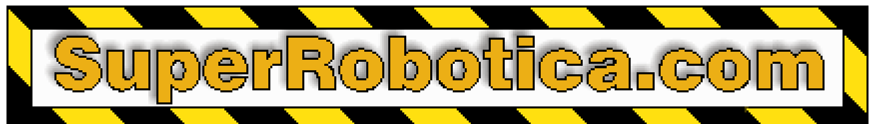
Ejemplo – supongamos que el fichero SerialPort.bas contiene las siguientes declaraciones de nivel de módulo:

```
Private Const InputBufferSize As Integer = 32
Private InputBuffer(1 To InputBufferSize) As Byte
```

EL fichero BXM contiene esta línea:

```
"serialport.inputbuffer",400,18,#FALSE#,1,1,32,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```





Valores decodificados:

```
       ModuleName:  "serialport"
     VariableName:  "inputbuffer"
   MemoryLocation:  400
         DataType:  Byte (18)
     IsPersistent:  False
       Dimensions:  1
    LowerBound(1):  1
    UpperBound(1):  32
 LowerBound(2..8):  0
 UpperBound(2..8):  0
```

Fichero MPX

Este fichero contiene información acerca de las variables del nivel de módulo en formato de código fuente. La función principal del fichero MPX es proporcionar información a un nodo remoto en la red para que el nodo remoto pueda obtener acceso de escritura/lectura a las variables estáticas del nodo local.

En el ejemplo anterior de puerto serie, el fichero MPX contiene las siguientes líneas pertenecientes a la matriz InputBuffer:

```
' serialport_inputbuffer data type: Byte
Public Const serialport_inputbuffer As Integer = 400 ' &H190
Public Const serialport_inputbuffer_IsPersistent As Boolean = False
```

Puede incluir el fichero MPX directamente como un módulo en el programa remoto. Cuando cambie el programa local, se actualizará automáticamente el fichero MPX al compilar el programa, por lo que puede recompilar el programa remoto para actualizarlo.

Nota – la dirección de cada variable tiene un rango numérico de 0 a 65.535. Sin embargo, el fichero MPX utiliza tipos de enteros con signo de 16 bits en los que se almacenan estos números. Mientras que las direcciones utilizan el rango 32 768 y 65 535 y se representan como número negativos con formato de complemento a dos. Los patrones de los bits internos corresponden a las direcciones reales.

Como ejemplo - la dirección aparente 1, corresponde con la dirección real 65 535 (&HFFFF).

Fichero OBJ

Estos ficheros contienen el código objeto compilado de cada subprograma. Los ficheros están escritos en el subdirectorio <Install>\ Temp, en el que <Install> es donde el programa BasicX.exe está instalado.

Una vez que se ha terminado de compilar un programa de BasicX, ya no serán necesarios los ficheros OBJ. Puede eliminar todos los ficheros del directorio Temp cuando el compilador haya acabado su trabajo.

Advertencia – no debería ejecutar 2 o más copias del compilador al mismo tiempo. Esto podría ocasionar que los compiladores interfieran el uno con el otro, dado que todos los ficheros de objetos se escriben en un directorio común.



Operación de la línea de comando

El compilador y el programa de descarga pueden activarse desde una línea de comandos. Esto es útil cuando está utilizando un editor de otros fabricantes para escribir su código BasicX. La sintaxis es la siguiente:

```
BasicX Project /c /d
```

BasicX – llama al fichero BasicX.exe. Si nada sigue a este parámetro, el entorno BasicX se iniciará sin un proyecto. La ruta de acceso puede también ser necesaria si el directorio actual es diferente al directorio de instalación.

Project (opcional) -- el nombre del fichero de su proyecto .BXP pero sin la extensión .BXP. Debe establecer el directorio de inicio donde esté ubicado este fichero. Si nada sigue a este parámetro, el entorno BasicX se iniciará con el proyecto especificado y cargado en el editor.

/c (opcional) -- indicador para compilar el proyecto. Esto creará el fichero .BXB para su proyecto. El entorno BasicX por si mismo no se iniciará cuando esta sentencia esté utilizada.

/d (opcional) -- indicador para descargar el proyecto. Esta sentencia (switch) descargará el fichero .BXB en su chip. EL fichero .BXB debe existir ya si se va a utilizar esta opción sin el indicador /c. El entorno BasicX por si mismo no se iniciará cuando esta sentencia esté utilizada.

La línea de comandos no es sensible al uso de mayúsculas / minúsculas. Cada argumento debe estar separado por 1 o más espacios o barras.

Deberá crear completamente el proyecto como siempre dentro del entorno BasicX. Una vez que se han establecido y guardado los nombres de los ficheros fuentes y preferencias del chip en el proyecto, podrá editar sus ficheros de códigos fuente en otro editor ASCII y utilizar la línea de comandos para compilar y descargar en proyecto. El directorio de inicio, el tipo de procesador (BX-01, BX-24, BX-35, etc.) y el puerto de descarga deben estar definidos correctamente para el proyecto actual.

Se creará un fichero un mensaje llamado *BasicX.err* en el directorio de inicio cuando utilice la opción de la línea de comandos. El fichero contiene los resultados del procesamiento. Si se encuentra algún error, aparecerá el número y descripción del error junto con los errores de tiempo de compilación, que generalmente incluye el número de línea y nombre de fichero del fichero fuente que causa el error.

Ejemplo – en el siguiente ejemplo, el fichero DemoProgram.bxp está ubicado en el directorio de inicio:

```
C:\Program Files\BasicX\>BasicX DemoProgram /c /d
```

Esta línea hace que el programa se compile y se descargue. Tenga en cuenta que se supone BasicX.exe está ubicado en el subdirectorio \Program Files\BasicX en este ejemplo. Esta ubicación puede variar dependiendo de dónde esté instalado el programa.